Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт информационных технологий и анализа данных

Центр программной инженерии

ОТЧЕТ

к лабораторной работе №3 по дисциплине:

|  |
| --- |
| «Объектно-ориентированное программирование» |
| Работа с потоками данных и обработка исключений |

наименование темы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Выполнил студент группы: | ИСТб-19-2 |  |  |  | Комогорцева Ю.В. |
|  | шифр группы |  | подпись |  | Фамилия ИО |
| Проверил: | доцент |  |  |  | Маланова Т.В. |
|  | должность |  | подпись |  | Фамилия ИО |

Иркутск. 2020 г.

Оглавление

[Оглавление 2](#_Toc56728292)

[1 Постановка задачи 3](#_Toc56728293)

[2 Описание структуры пользовательского меню 4](#_Toc56728294)

[3 Проектирование классов 5](#_Toc56728295)

[4 Описание методов классов и спецификация локальных переменных 6](#_Toc56728296)

[5 Таблица тестов 10](#_Toc56728297)

[6 Результаты тестирования 16](#_Toc56728298)

[7 Код программы 20](#_Toc56728299)

[Список использованных источников 29](#_Toc56728300)

1 Постановка задачи

1. Применение на практике объектно-ориентированного подхода для организации системы ввода/вывода.

2. Получение практического опыта описания и реализации классов, создания объектов, вызова методов, а также навыков работы с потоками данных и обработки исключений на базе платформы Java.

Вариант 11

1. Выберите из предложения слово, которое располагается в центре предложения, то есть символ, входящий в его состав, является центральным символом в предложении (с учетом пробелов).
2. Найдите символы в строке, который встречается менее 10%.
3. Найдите длину первого слова в предложении, а затем выберите из предложения все слова той же длины.

2 Описание структуры пользовательского меню

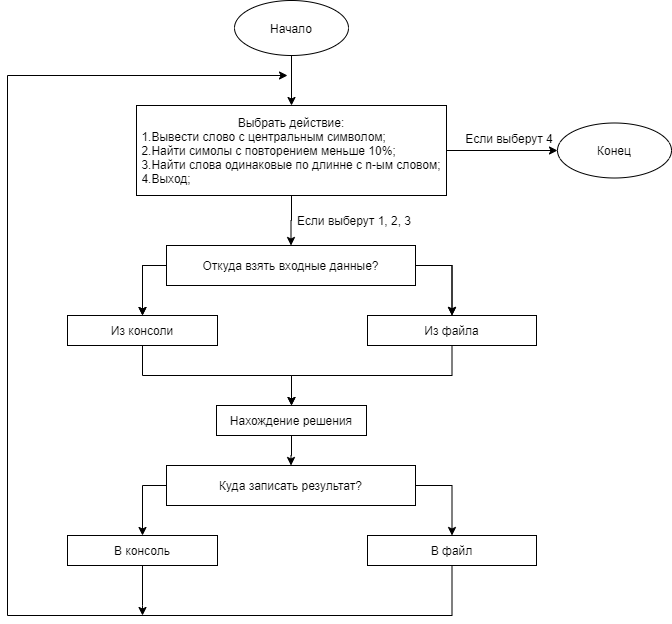


Рисунок 1 – Описание структуры пользовательского меню

3 Проектирование классов

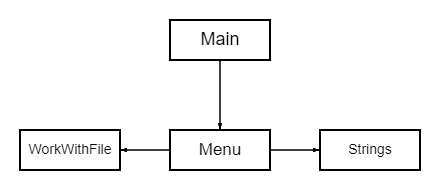


Рисунок 2 – Проектирование классов

4 Описание методов классов и спецификация локальных переменных

Таблица 1

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Имя** | **Описание** | **Тип данных** |
| **public class Main** | | | |
| **public static void main(String[] args)** | | | |
| **class Strings** | | | |
| 1 | S | Глобальная переменная класса для удобной работы с методами | String |
| **public static int GetCenter()** | | | |
| 2 | center | Центральный индекс | int |
| **public static String getResult1(String S)** | | | |
| 3 | center | Центральный индекс | int |
| 4 | results | Центральное слово/символ | String |
| 5 | rez | Строка с итоговым результатом | String |
| **public static String getResult2(String S)** | | | |
| 6 | mas | Преобразованная строка в массив символов | char[] |
| 7 | result | Массив для результата с итоговыми символами | char[] |
| 8 | t | Переменная отвечающая за наличие повторяющихся символов | int |
| 9 | i,j | Переменные для цикла | int |
| 10 | number | Счетчик количества повторений какого либо символа | double |
| 11 | rez | Строка с итоговым результатом | String |
| **public static String getResult3(String S)** | | | |
| 12 | str | Массив слов взятых из строки | String[] |
| 13 | length | Длинна первого слова | int |
| 14 | t | Счетчик количества искомых слов | int |
| 15 | i | Переменная для цикла | int |
| 16 | rez | Строка с итоговым результатом | String |
| **public class Menu-**класс реализующий меню | | | |
| 17 | in | Объект класса Scanner для ввода данных | Scanner |
| **public static void mainMenu()-**метод для главного меню с выбором действий | | | |
| 18 | key1 | Переменная для номера действия | int |
| **public static void MakeTask1()**-выполняет задание 1 | | | |
| 19 | keyInput | Переменная для номера типа ввода информации (консоль-1;файл-2) | int |
| 20 | String1 | Обьект класса Strings , необходимый для решения задачи | Strings |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| **public static void MakeTask2()-**выполняет задание 2 | | | |
| 21 | keyInput | Переменная для номера типа ввода информации(консоль-1;файл-2) | int |
| 22 | String2 | Обьект класса Strings ,необходимый для решения задачи | Strings |
| **public static void MakeTask3()-**выполняет задание 3 | | | |
| 23 | keyInput | Переменная для номера типа ввода информации(консоль-1;файл-2) | int |
| 24 | String3 | Обьект класса Strings ,необходимый для решения задачи | Strings |
| 25 | NumberWord | Переменная для номера слова с длинной которого будем сравнивать | int |
| **public static int InputInt()**-метод для проверки что в консоль ввели целое число | | | |
| 26 | read | Переменная для считывания с консоли | int |
| **public static String InputStr()**-метод для проверки что в консоль ввели строку | | | |
| 27 | rez | Переменная со строкой которую ввел пользователь | String |
| 28 | i | Переменная , что бы проверить ,что пользователь ввел что-то | int |
| **public static void OutputData(String rezult)**-метод для выбора места и вывода результата в это место | | | |
| 29 | keyOutput | Переменная для номера типа вывода информации(консоль-1;файл-2) | int |
| 30 | rezult | Строка с результатом которую нужно вывести | String |
| **public static void printMenuInput()**-вывод в консоль содержание меню ввода информации | | | |
| **public class WorkWithFile-**класс для работы с файлами | | | |
| **public static String strFromTxtFile(String filename)**-метод для чтения информации из файла | | | |
| 31 | filename | Путь файла для чтения данных | String |
| 32 | rez |  | String |
| 33 | sb |  | StringBuffer |
| 34 | br |  | BufferedReader |
| **public static void strToTxtFile(String str, String filename)-** метод для записи информации в файл | | | |
| 35 | br |  | BufferedWriter |
| 36 | str | Строка с результатом которую нужно записать | String |
| 37 | filename | Путь файла для записи данных | String |

**Сlass Strings**

1. private String S;
2. public Strings(String S)ꟷконструктор с параметрами
3. public Strings()ꟷ конструктор по умолчанию
4. public Strings(Strings String) ꟷ конструктор копирования
5. public String getS()ꟷВозвращение атрибута
6. public void setS(String s) изменение атрибута
7. public static int GetCenter(String S) ꟷметод возвращающий центральный индекс в строке
8. public static String getResult1(String S) ꟷметод выполняющий задание 1, находящий слово которое содержит центральный символ в строке
9. public static String getResult2(String S) ꟷ метод выполняющий задание 2, находит символы которые повторяются меньше 10%
10. public static String getResult3(String S) ꟷметод выполняющий задание 3, находит слова одинаковые по длинне с первым словом

**public class WorkWithFile(класс для работы с файлами)**

1. public static String strFromTxtFile(String filename)-метод для чтения информации из файла
2. public static void strToTxtFile(String str, String filename)- метод для записи информации в файл

**public class Menu(класс реализующий меню)**

1. public static void mainMenu()-метод для главного меню с выбором действий
2. public static void MakeTask1()-выполняет задание 1
3. public static void MakeTask2()-выполняет задание 2
4. public static void MakeTask3()-выполняет задание 3
5. public static String InputStr()-метод для проверки что в консоль ввели строку
6. public static int InputInt()-метод для проверки что в консоль ввели целое число
7. public static void OutputData(String rezult)-метод для выбора места и вывода результата в это место
8. public static void printMenuInput()-вывод в консоль содержание меню ввода информации

5 Таблица тестов

Таблица 2

|  |  |  |
| --- | --- | --- |
| **№** | **Входные данные** | **Данные на выходе** |
| **Первая задача** | | |
| 1 | 1  1  I am not young enough to know everything.  1  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 1  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  1  Введите предложение:I am not young enough to know everything.  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  1  Центральное слово: enough  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 2 | 1  2  2  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 1  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  2  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  2  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| **Вторая задача** | | |
| 3 | 2  1  beautiful beach  2  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 2  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  1  Введите предложение:beautiful beach  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  2  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 4 | 2  2  1  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 2  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  2  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  1  Нет повторяющихся символов меньше 10%  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| **Третья задача** | | |
| 5 | 3  1  Life is like riding a bicycle . To keep your balance you must keep moving .  100  1  1  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 3  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  1  Введите предложение:Life is like riding a bicycle . To keep your balance you must keep moving .  С длинной какого слова будем сравнивать?:100  Слова с таким номером не существует,введите еще раз1  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  1  Слова такой же длинны как и слово Life : like, keep, your, must, keep,  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 6 | 3  1  tomorrow is another day  1  2  4 | Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 3  Выберите, откуда взять информацию:  1.Ввод с консоли;  2.Ввод из файла;  1  Введите предложение:tomorrow is another day  С длинной какого слова будем сравнивать?:1  Выберите, куда вывести результат:  1.В консоль;  2.В текстовый файл;  2  Выберите:  1.Вывести слово с центральным символом;  2.Найти симолы с повторением меньше 10%;  3.Найти слова одинаковые по длинне с n-ым словом;  4.Выход;  Введите номер : 4  До скорой встречи! |

6 Результаты тестирования

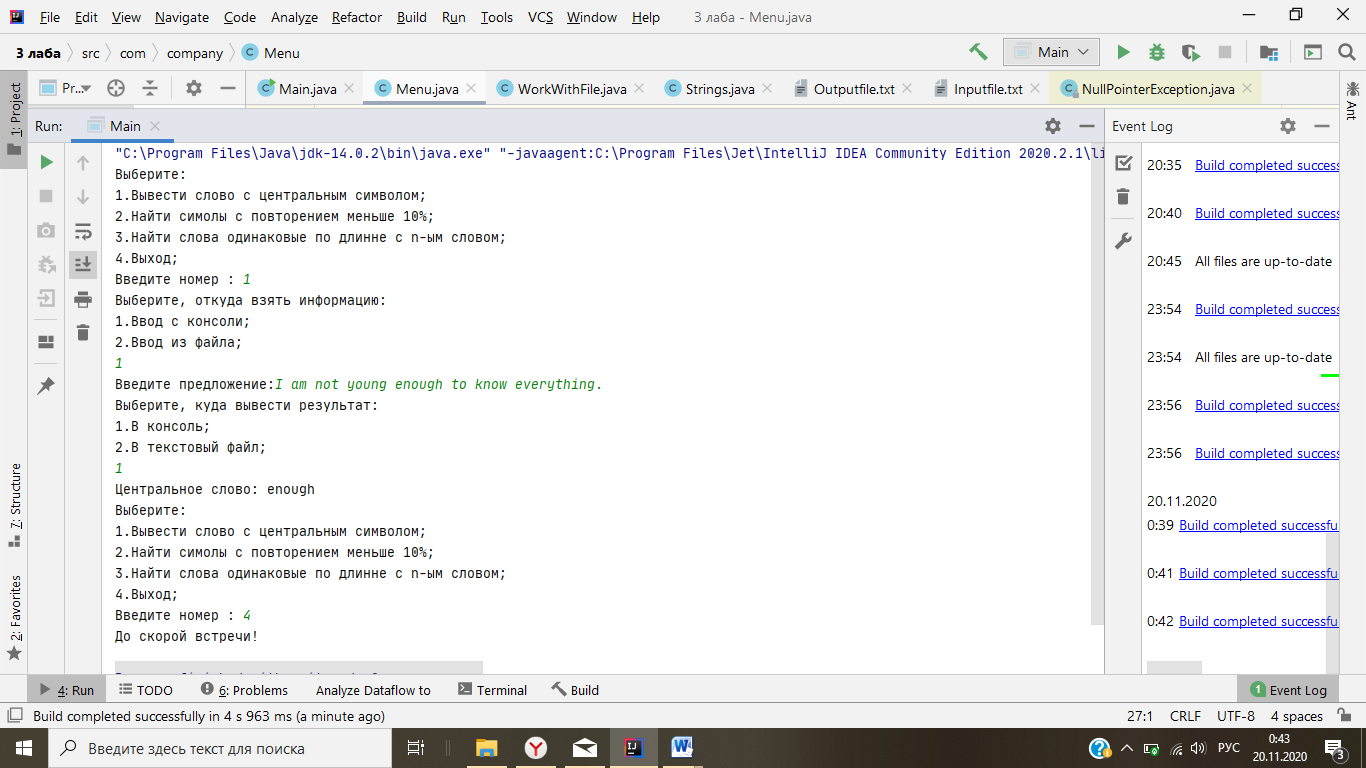


Рисунок 3 – Тест 1

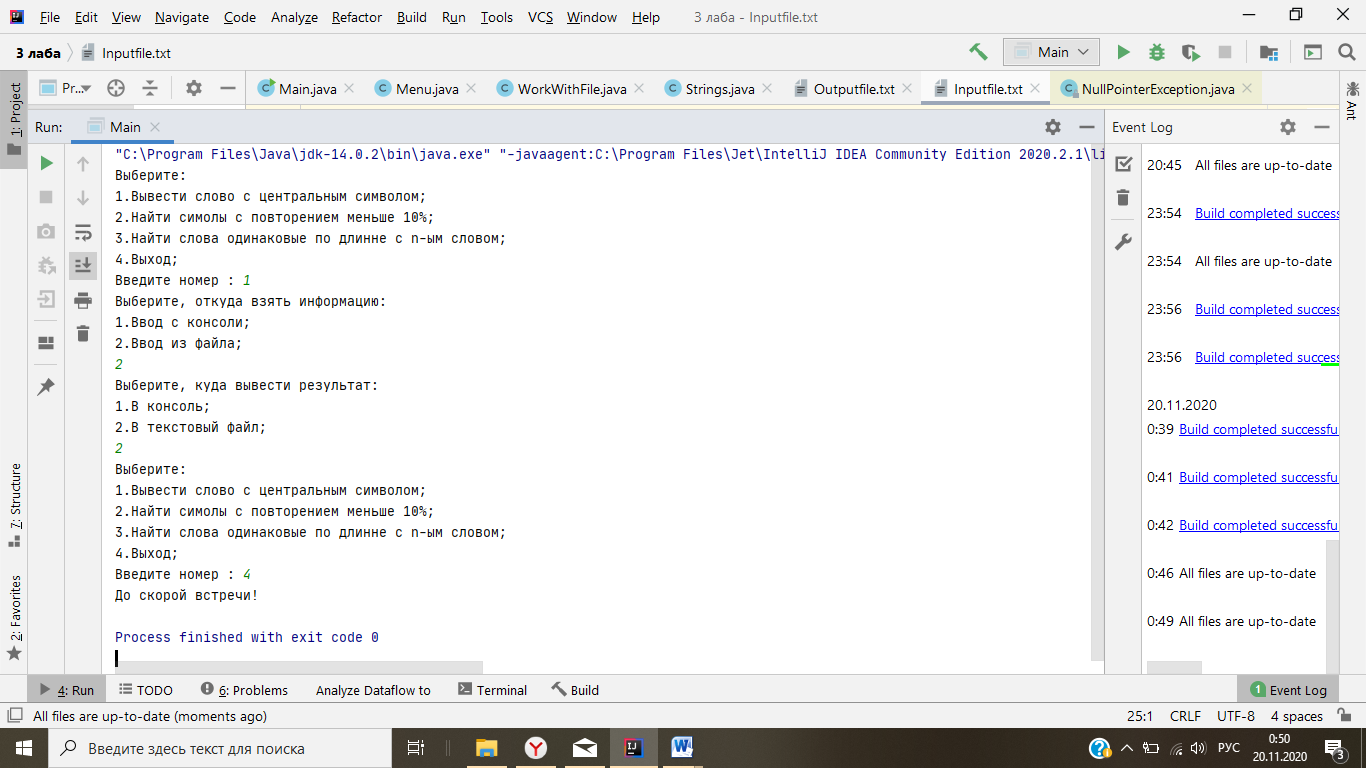


Рисунок 4 – Тест 2

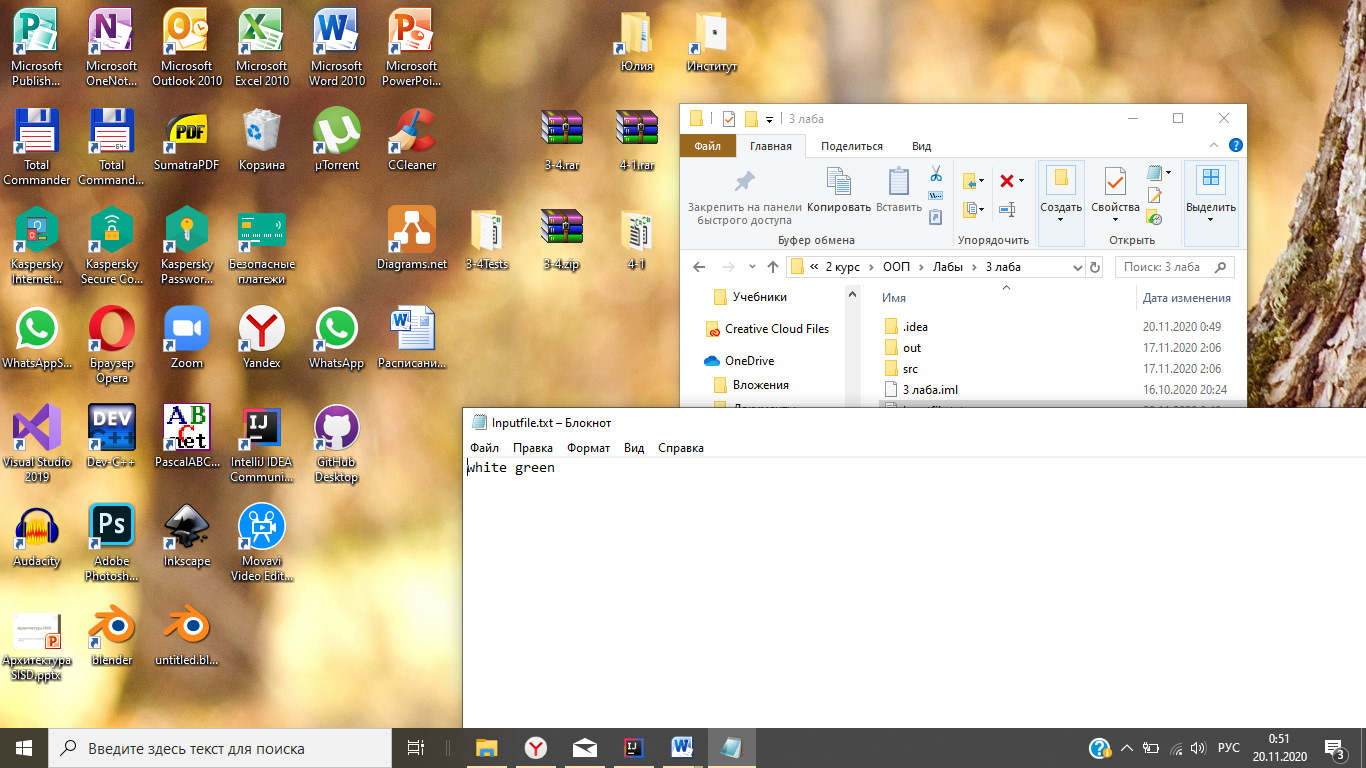


Рисунок 5 – Тест 2

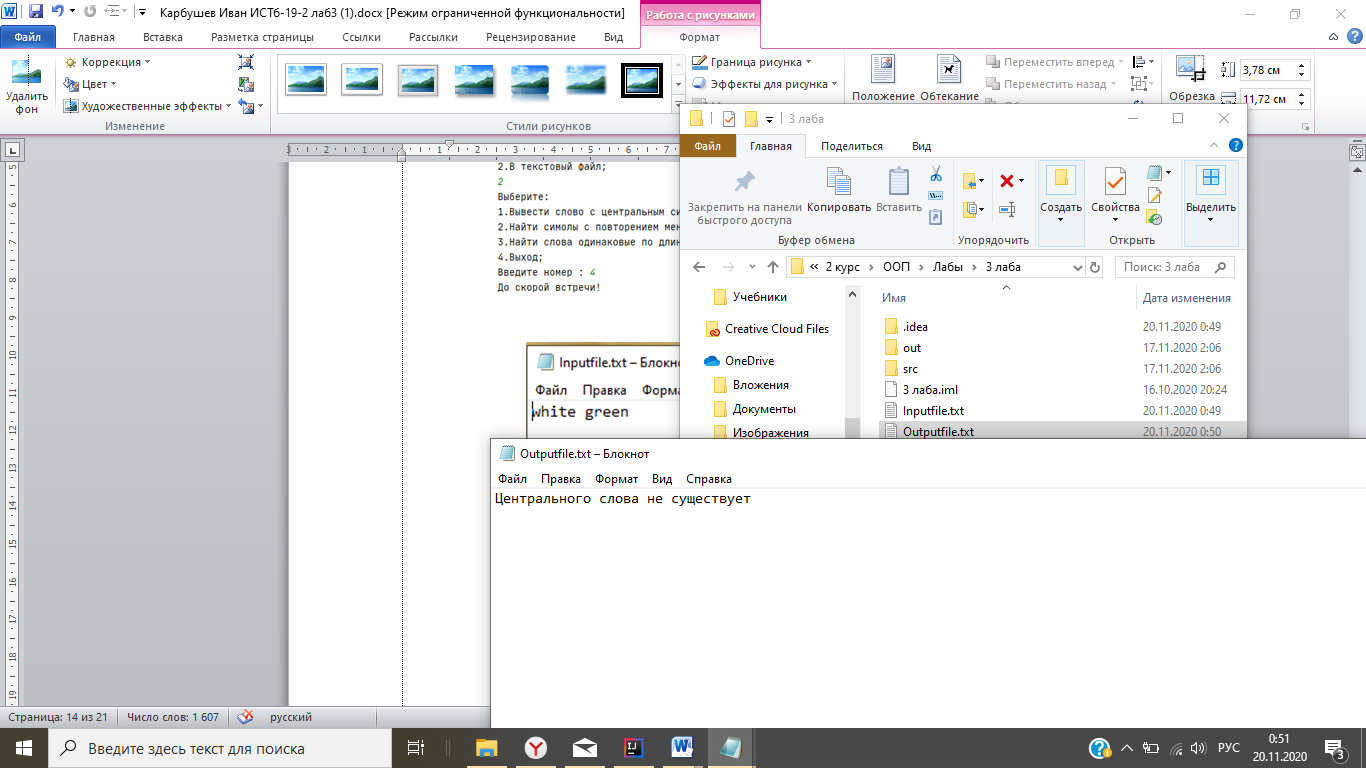


Рисунок 6 – Тест 2

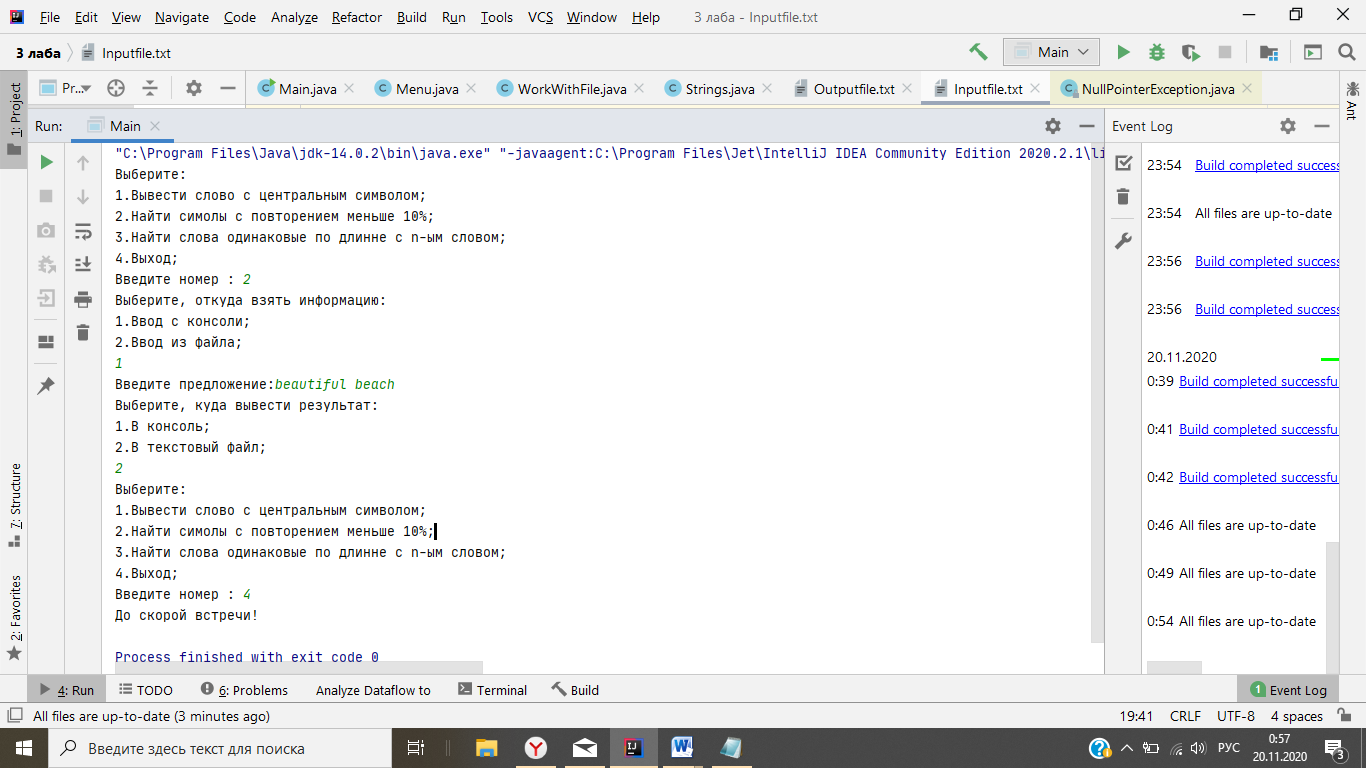


Рисунок 7 – Тест 3

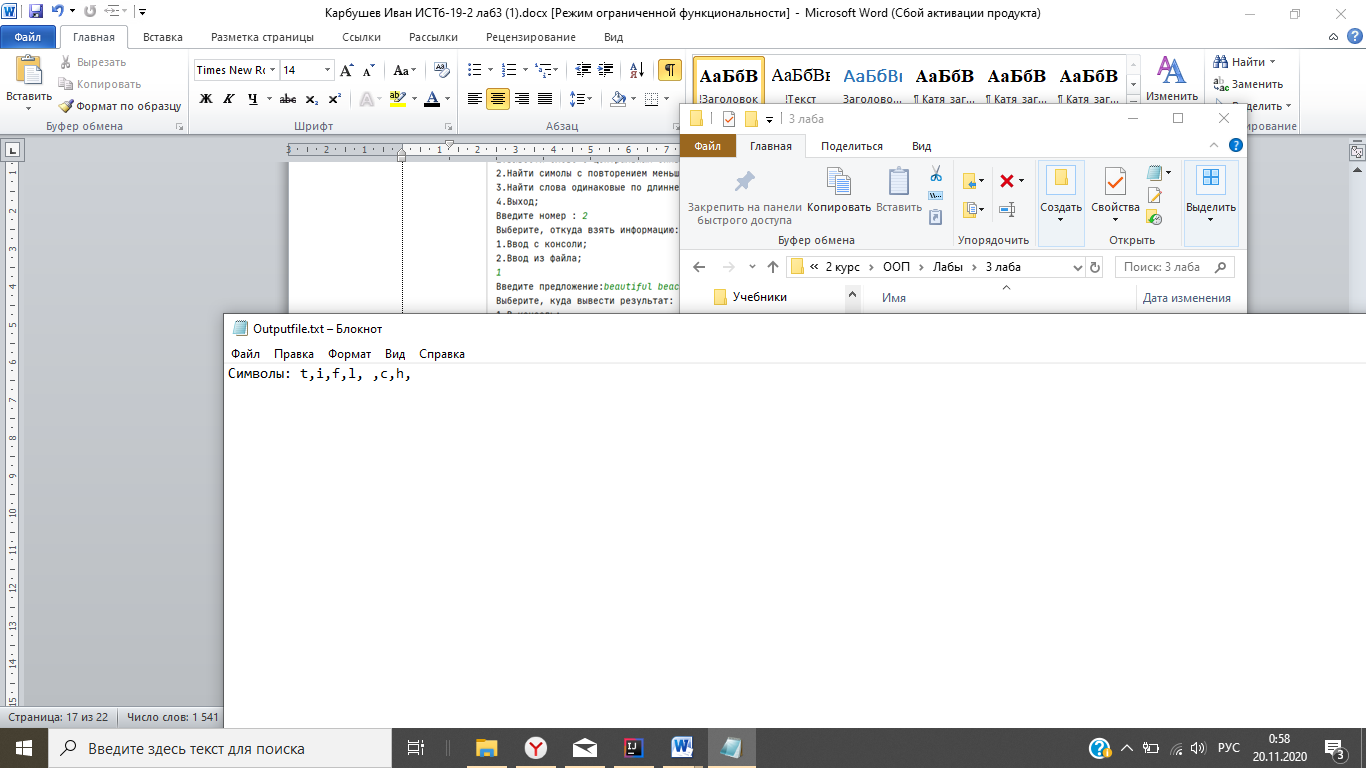


Рисунок 8 – Тест 3

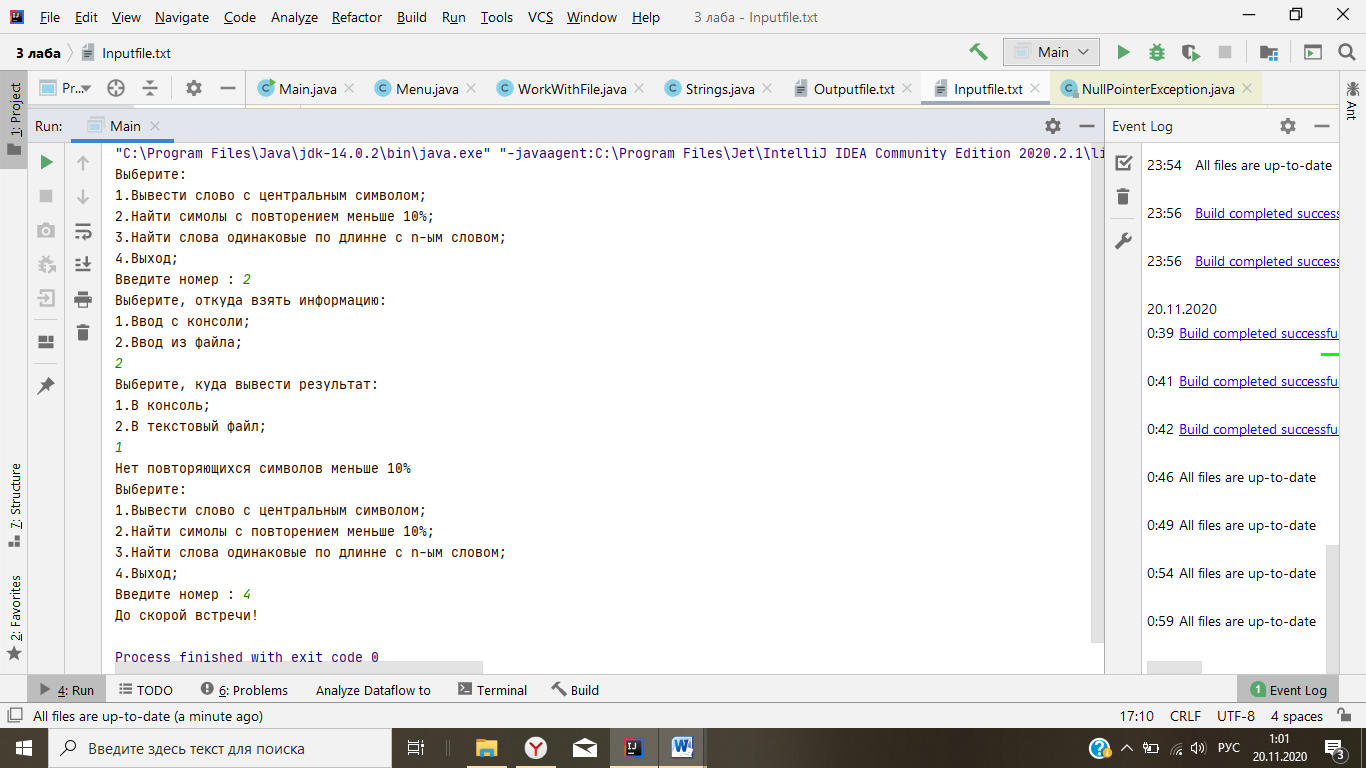


Рисунок 9 – Тест 4

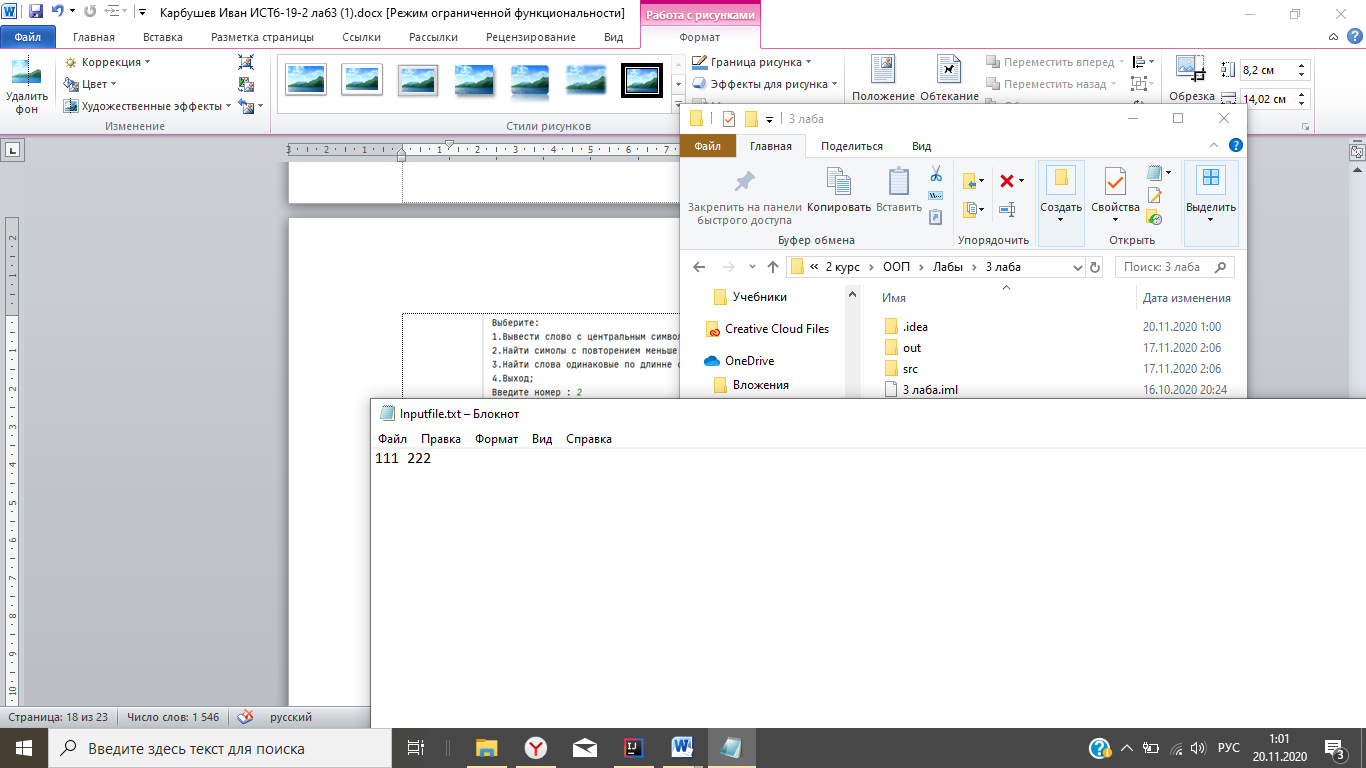


Рисунок 10 – Тест 4

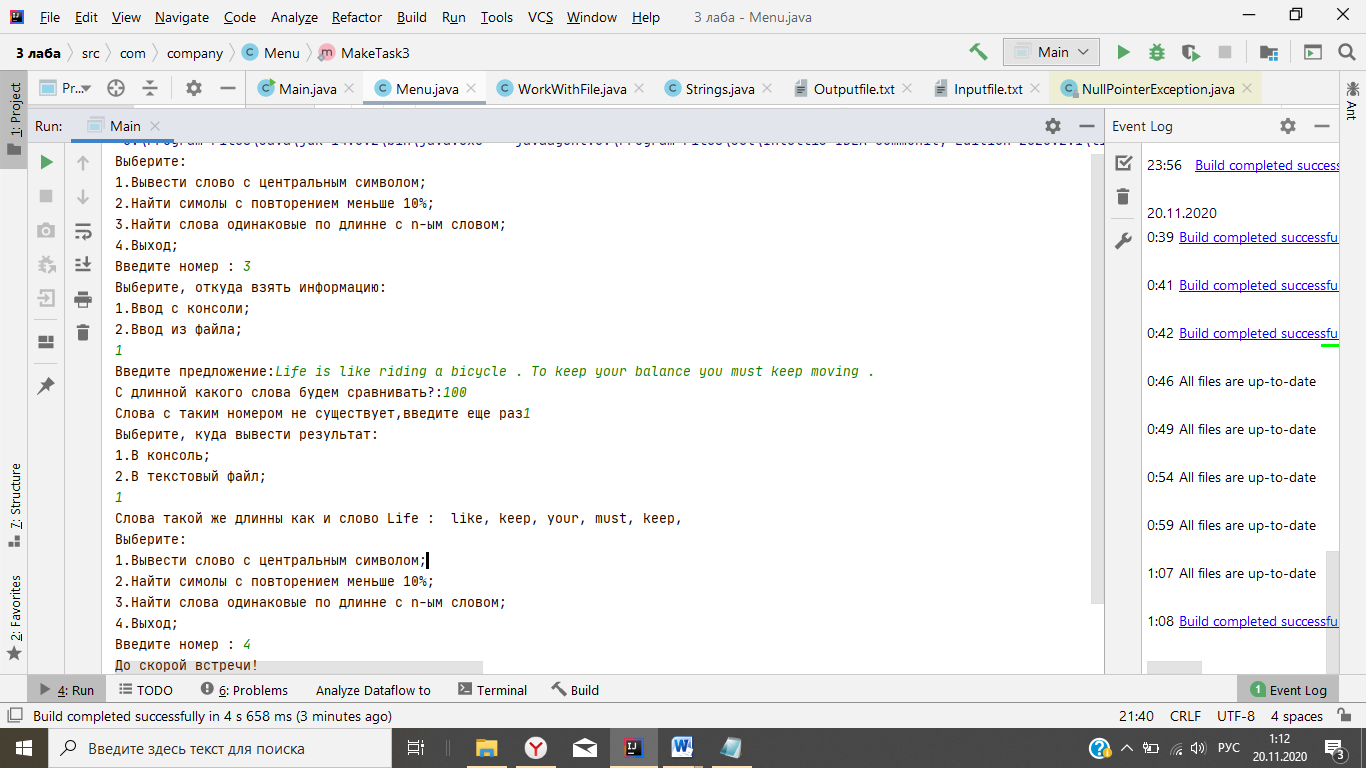


Рисунок 11 – Тест 5

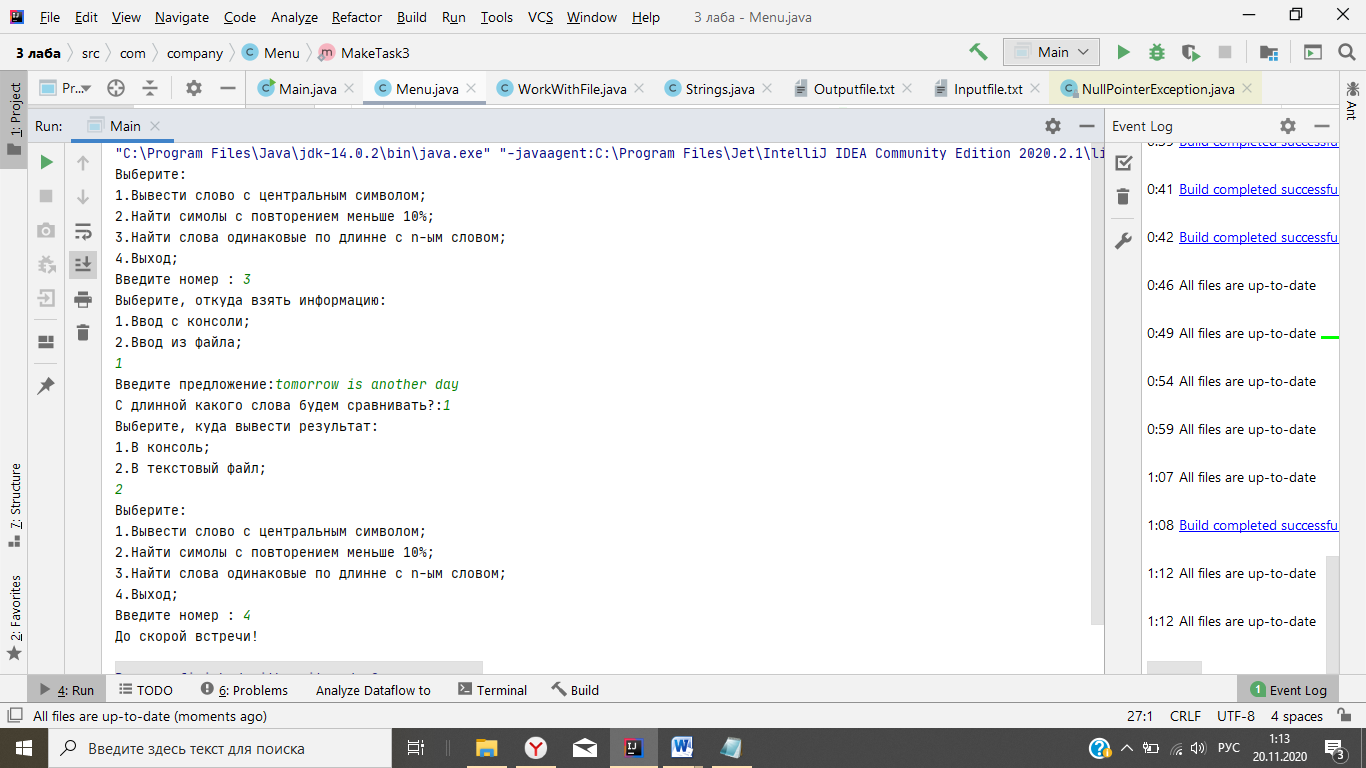


Рисунок 12 – Тест 6

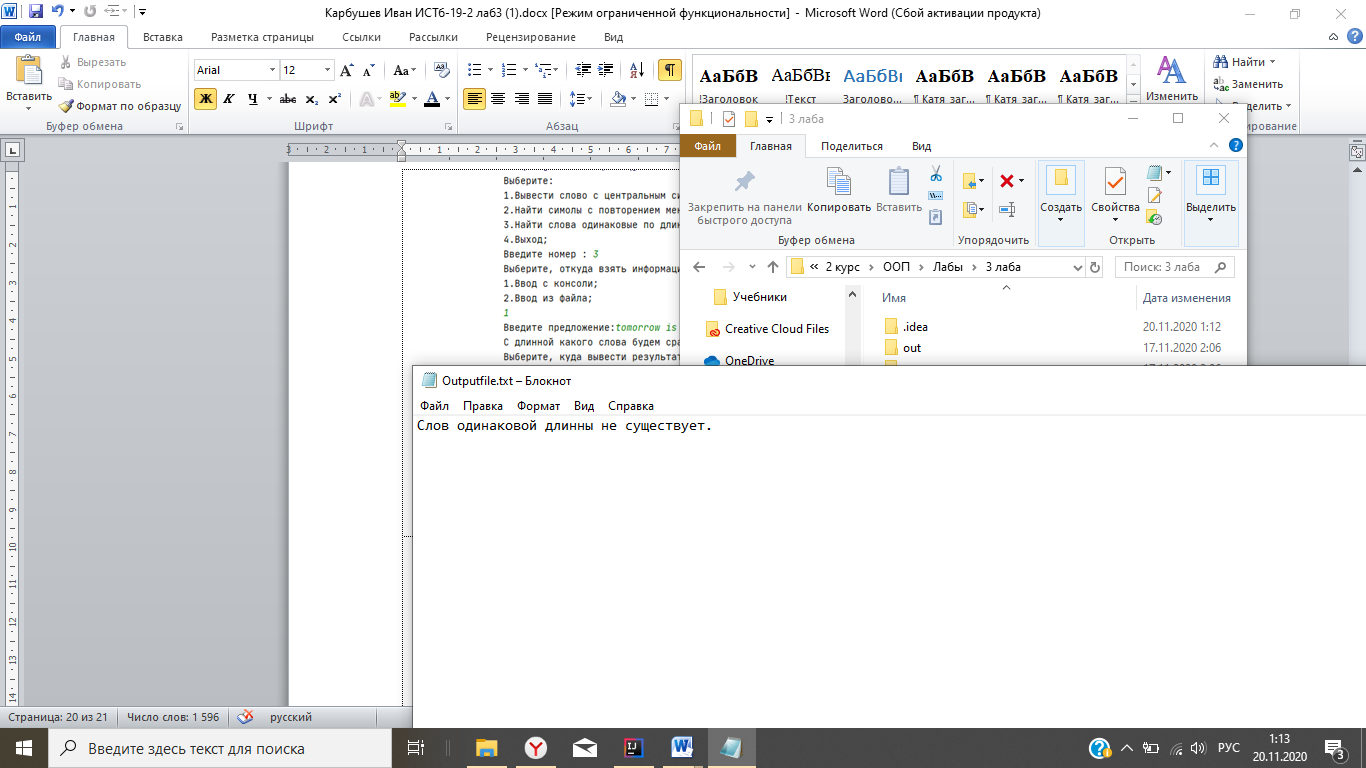


Рисунок 13 – Тест 6

7 Код программы

**class Main**

**package** com.company;  
**public class** Main {  
 **public static void** main(String[] args) {  
 Menu.*mainMenu*();  
 }  
}

**class Menu**

**package** com.company;  
**import** java.util.Scanner;  
  
**public class** Menu {  
 **public static** Scanner *in* = **new** Scanner(System.***in***);  
  
 **public static void** mainMenu() {  
 **int** key1=0;  
 **do** {  
 System.***out***.println(**"Выберите:"**);  
 System.***out***.println(**"1.Вывести слово с центральным символом;"**);  
 System.***out***.println(**"2.Найти симолы с повторением меньше 10%;"**);  
 System.***out***.println(**"3.Найти слова одинаковые по длинне с n-ым словом;"**);  
 System.***out***.println(**"4.Выход;"**);  
 System.***out***.print(**"Введите номер : "**);  
 *//ввели и проверили что бы это было число* key1=*InputInt*();  
 **switch** (key1) {  
 **case** 1:  
 *//задание1  
 MakeTask1*();  
 **break**;  
 **case** 2:  
 *//задание2  
 MakeTask2*();  
 **break**;  
 **case** 3:  
 *//задание3  
 MakeTask3*();  
 **break**;  
 **case** 4:  
 *//выход* **break**;  
 **default**:  
 *//если введут не 1,2,3 или 4 ,то такого пункта нет и заново попросит ввести  
 //а не выйдет из программы потому что ключ будет не равен 4* System.***out***.println(**"Нет такого пункта"**);  
 }  
 } **while** (key1!=4);  
 System.***out***.println(**"До скорой встречи!"**);  
 }  
  
 *//делает задание 1* **public static void** MakeTask1() {  
 *//выводит меню ввода  
 printMenuInput*();  
 *//читает и проверят на число* **int** keyInput = *InputInt*();  
 *//создаем обьект,необходимый для решения* Strings String1=**new** Strings();  
 **switch** (keyInput) {  
 **case** 1:  
 *//ввод с консоли* System.***out***.print(**"Введите предложение:"**);  
 String1.setS(*InputStr*());  
 *//вывод результата  
 OutputData*(Strings.*getResult1*(String1.getS()));  
 **break**;  
 **case** 2:  
 *//ввод с файла  
 //если файл не пустой то считать данные оттуда и найти и вывести результат* **if**(WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**)!=**null**){  
 String1.setS(WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**));  
 *OutputData*(Strings.*getResult1*(String1.getS()));  
 }  
 **break**;  
 **default**:  
 System.***out***.println(**"Нет такого пункта"**);  
 }  
 }  
  
 *//делает задание 2* **public static void** MakeTask2() {  
 *printMenuInput*();  
 **int** keyInput = *InputInt*();  
 Strings String2=**new** Strings();  
 **switch** (keyInput) {  
 **case** 1:  
 *//ввод с консоли* System.***out***.print(**"Введите предложение:"**);  
 String2.setS(*InputStr*());  
 *//вывод результата  
 OutputData*(Strings.*getResult2*(String2.getS()));  
 **break**;  
 **case** 2:  
 *//ввод с файла  
 //если файл не пустой то считать данные оттуда и найти и вывести результат* **if**(WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**)!=**null**){  
 String2.setS(WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**));  
 *OutputData*(Strings.*getResult2*(String2.getS()));  
 }  
 **break**;  
 **default**:  
 System.***out***.println(**"Нет такого пункта"**);  
 }  
 }  
  
 *//делает задание 3* **public static void** MakeTask3() {  
 *printMenuInput*();  
 **int** NumberWord=0;  
 Strings String3=**new** Strings();  
 **int** keyInput = *InputInt*();  
 **switch** (keyInput) {  
 **case** 1:  
 *//ввод с консоли* System.***out***.print(**"Введите предложение:"**);  
 String3.setS(*InputStr*());  
 System.***out***.print(**"С длинной какого слова будем сравнивать?:"**);  
 *//проверка на число* NumberWord=*InputInt*()-1;  
 *//массив слов из файла  
 //если номер слова из файла больше чем количество слов или меньше* **while** ((NumberWord> String3.getS().split(**" "**).**length**)||((NumberWord<0))){  
 System.***out***.print(**"Слова с таким номером не существует,введите еще раз"**);  
 NumberWord=*InputInt*()-1;  
 }  
 *//вывод результата  
 OutputData*(Strings.*getResult3*(String3.getS(), NumberWord));  
 **break**;  
 **case** 2:  
 *//ввод с файла  
 //если файл не пустой то считать данные оттуда и найти и вывести результат* **if**(WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**)!=**null**) {  
 String s = WorkWithFile.*strFromTxtFile*(**"Inputfile.txt"**);  
 *//в обьект записываем данные из файла до черты |* String3.setS(s.substring(0, s.indexOf(**"|"**)));  
 *//массив из слов в предложение  
 //длинна всей строки* **int** t = s.length();  
 *//номер слова из файла* NumberWord = Integer.*parseInt*(s.substring(s.indexOf(**"|"**) + 1, t)) - 1;  
 *//если номер слова из файла больше чем количество слов или меньше* **if** ((NumberWord > String3.getS().split(**" "**).**length**) || (NumberWord < 0)) {  
 *OutputData*(**"Слова с таким номером не существует"**);  
 } **else** {  
 *//вывод результата  
 OutputData*(Strings.*getResult3*(String3.getS(), NumberWord));  
 }  
 }  
 **break**;  
 **default**:  
 System.***out***.println(**"Нет такого пункта"**);  
 }  
 }  
  
 *//метод для проверки что в консоли ввели число* **public static int** InputInt(){  
 **int** read ;  
 **try**{  
 read = Integer.*parseInt*(*in*.nextLine());  
 }**catch** (NumberFormatException ex){  
 System.***out***.print(**"Вводите только числа,пожалуйста"**);  
 read=*InputInt*();  
 }  
 **return** read;  
 }  
  
 *//метод для проверки что в консоли ввели строку* **public static** String InputStr(){  
 String rez=**null**;  
 **try**{  
 rez=*in*.nextLine();  
 **int** i=1/rez.length();  
 }  
 **catch** (ArithmeticException e) {  
 System.***out***.println(**"Вы ничего не ввели.Повторите ввод:"**);  
 *InputStr*();  
 }  
 **return** rez;  
 }  
  
 *//вывод результата* **public static void** OutputData(String rezult) {  
 System.***out***.println(**"Выберите, куда вывести результат:"**);  
 System.***out***.println(**"1.В консоль;"**);  
 System.***out***.println(**"2.В текстовый файл;"**);  
 *//опять ввели и проверили число ли это* **int** keyOutput=*InputInt*();  
 **switch** (keyOutput){  
 **case** 1:  
 *//вывод в консоль* System.***out***.println(rezult);  
 **break**;  
 **case** 2:  
 *//вывод в файл* WorkWithFile.*strToTxtFile*(rezult,**"Outputfile.txt"**);  
 **break**;  
 **default**:  
 *//если ввели не 1 и 2 ,то попросит ввести еще раз* System.***out***.println(**"Нет такого пункта,повторите ввод"**);  
 *OutputData*(rezult);  
 }  
 }  
  
 *//вывод меню ввода на консоль* **public static void** printMenuInput() {  
 System.***out***.println(**"Выберите, откуда взять информацию:"**);  
 System.***out***.println(**"1.Ввод с консоли;"**);  
 System.***out***.println(**"2.Ввод из файла;"**);  
  
 }  
}

**class WorkWithFile**

**package** com.company;  
**import** java.io.\*;  
**public class** WorkWithFile {  
 *//чтение из файла* **public static** String strFromTxtFile(String filename){  
 String rez;  
 **try**(BufferedReader br=**new** BufferedReader(**new** FileReader(filename))){  
 StringBuffer sb = **new** StringBuffer();  
 rez=br.readLine();  
 **for** (**int** i=0;i<rez.length();i++){  
 sb.append(rez.charAt(i));  
 }  
 rez=sb.toString();  
 }  
 **catch** (NullPointerException e) {  
 System.***out***.println(**"Файл пустой"**);  
 rez=**null**;  
 }  
 **catch** (IOException e) {  
 System.***out***.println(**"Ошибка чтения"**);  
 rez=**null**;  
 }  
 **return** rez;  
 }  
  
 *//запись в файл* **public static void** strToTxtFile(String str, String filename){  
 **try**(BufferedWriter br=**new** BufferedWriter(**new** FileWriter(filename))){  
 br.write(str);  
 }  
 **catch** (IOException e) {  
 System.***out***.println(**"Ошибка записи"**);  
 }  
 }  
  
}

**class Strings**

**package** com.company;  
**class** Strings {  
 **private** String **S**;  
  
 **public** Strings(String S) { *//конструктор с параметрами* **this**.**S** = S;  
 }  
 **public** Strings() { *//конструктор по умолчанию* **this**.**S** = **null**;  
 }  
 **public** Strings(Strings S) { *//конструктор копирования* **this**.**S** = S.**S**;  
 }  
  
 **public** String getS() { *//Возвращение атрибута* **return S**;  
 }  
 **public void** setS(String s) { *//изменение атрибута* **this**.**S** = s;  
 }  
 **public static int** GetCenter(String S){ *//метод возвращающий центральный индекс в строке* **int** center;  
 **if**(S.length()%2==0) *//Если длинна строки четная то* center =(S.length()/2)-1; *//центральный индекс равен половине длинне строки-1* **else** *//иначе* center =(S.length()/2); *//центральный индекс равен половине длинне строки* **return** center;  
 }  
 **public static** String getResult1(String S) { *//метод выполняющий задание 1* **int** center = *GetCenter*(S);*//нахождение центрального индекса* String[] arr = S.split(**" "**);  
 String rez=**null**;  
 *//нахождение слова из строки от центрального символа влево и вправо до пробела* String results = S.substring(S.lastIndexOf(**" "**, center), S.indexOf(**" "**, center));  
 **if** (!results.contains(**" "**)) { *//если центральный символ равен пробелу* rez = **"Центрального слова не существует"**;*//то центрального слова не существует* } **else** { *//иначе* rez = **"Центральное слово:"** + results; *//центральное слово найдено* }  
 **return** rez;  
 }  
 **public static** String getResult2(String S){ *//метод выполняющий задание 2* **char**[] mas = S.toCharArray(); *//преобразование строки в массив символов* **char**[] result = **new char**[S.length()]; *//создание массива для итоговых символов* **int** t = 0; *//переменная отвечающая за наличие повторяющихся символов* **for** (**int** i = 0; i < S.length(); i++) { *//цикл на перебор символов в строке* **double** number = 0; *//переменная количества повторяющейся буквы* **for** (**int** j = 0; j < S.length(); j++) { *//цикл на перебор символов в строке* **if** (mas[i] == mas[j]) { *//если символы равны* number++; *//то количество повторений увеличивается на 1* }  
 }  
 **if** ((100 \* number / S.length()) < 10) *//если процент повторений меньше 10* {  
 result[t] = mas[i]; *//то символ записываем в массив с результатом* t = t + 1; *//увеличиваем наличие повторяющихся символов меньше 10%* }  
 }  
 String rez=**""**;  
 **if** (t == 0) { *//если нет повторяющихся символов меньше 10%* rez=**"Нет повторяющихся символов меньше 10%"**; *//то сообщаем об этом* }  
 **else** { *//иначе* rez=**"Символы: "**; *//вывод массива с повторяющимися символами меньше 10%* **for** (**int** i = 0; i < t; i++) {  
 rez=rez+result[i] + **","**;  
 }  
 }  
 **return** rez;  
 }  
 **public static** String getResult3(String S,**int** word) { *//метод выполняющий задание 3* String[] str = S.split(**" "**); *//создаем массив слов взятых из строки* **int** length = str[word].length(); *//длинна слова* String rez=**"Слова такой же длинны как и слово "**+str[word]+**" : "**;  
 **int** t=0; *//счетчик для наличия искомых слов* **for**(**int** i = 1; i < str.**length**; i++) *//Цикл перебора слов до окончания строки* {  
 **if** (length == str[i].length()) *//если длинна первого слова равна длине выбранного слова* {  
 rez=rez+str[i]+**", "**;  
 t=t+1; *//счетчик увеличивается* }  
 }  
 **if** (t==0) { *//если счетчик для наличия искомых слов равен 0* rez=**"Слов одинаковой длинны не существует."**; *//то слов одинаковой длинны не существует* }  
 **if** (word==-1) { *//если счетчик для наличия искомых слов равен 0* rez=**"Слова с таким номером не существует."**; *//то слов одинаковой длинны не существует* }  
 **return** rez;  
 }  
  
}

Список использованных источников

1. Объектно-ориентированное программирование. Метод. указания по

выполнению лабораторных работ / сост.: В.Л. Аршинский. – Иркутск : Изд-во

ИРНИТУ, 2017. – 24 c.